Resource Allocation using Machine learning

Cloud Computing (CSN-520)

Under the guidance of

Prof. Sateesh Kumar Peddoju

Submitted By

Rahul Kurkure (20114079), Sanchit Chaba (20114083)

Introduction

- In the realm of cloud computing, efficient resource allocation is pivotal for optimal performance and user satisfaction.
- Traditional methods struggle with the dynamic nature of cloud workloads.
- This project focuses on comparing two resource allocation approaches: Random Forest Regression (RFR) and Genetic Algorithms (GA).
  RFR employs ensemble learning, while GA mimics natural selection.
- Evaluation criteria encompass scalability, adaptability to changing workloads, computational efficiency, and overall performance optimization.

Problem Statement

The project aims to assess and compare the effectiveness of Random Forest Regression (RFR) and Genetic Algorithms (GA) for dynamic resource allocation in cloud computing. Key metrics include scalability, adaptability, computational efficiency, and performance optimization.

Dataset Used

5G Resource Allocation Dataset

https://www.kaggle.com/datasets/omarsobhy14/5g-quality-of-service

Timestamp Date/Time	User_ID Text	Application_Type Text	Signal_Strength Number	Latency Number	Require Number
2023-09-03 2023-09-03	400 unique values	Video_Call 14% Web_Browsing 12% Other (294) 74%	-85 dBm 2% -97 dBm 2% Other (382) 96%	5 ms 9% 30 ms 3% Other (353) 88%	0.1 Mbps 0.5 Mbps Other (37
9/3/2023 10:00	User_1	Video_Call	-75 dBm	30 ms	10 Mbps
9/3/2023 10:00	User_2	Voice_Call	-80 dBm	20 ms	100 Kbps
9/3/2023 10:00	User_3	Streaming	-85 dBm	40 ms	5 Mbps
9/3/2023 10:00	User_4	Emergency_Service	-70 dBm	10 ms	1 Mbps
9/3/2023 10:00	User_5	Online_Gaming	-78 dBm	25 ms	2 Mbps

Genetic Algorithm

Genetic Algorithm (GA) is a powerful optimization technique inspired by natural selection.

In cloud computing, GA aids in dynamically assigning virtual machines (VMs) to physical machines (PMs) based on workload demands. It operates through selection, crossover, and mutation, iteratively improving solutions' fitness. GA's adaptability enables it to explore large solution spaces and adjust to changing conditions.

By balancing exploration and exploitation, GA offers an efficient approach to resource allocation, optimizing performance and resource utilization in cloud environments.

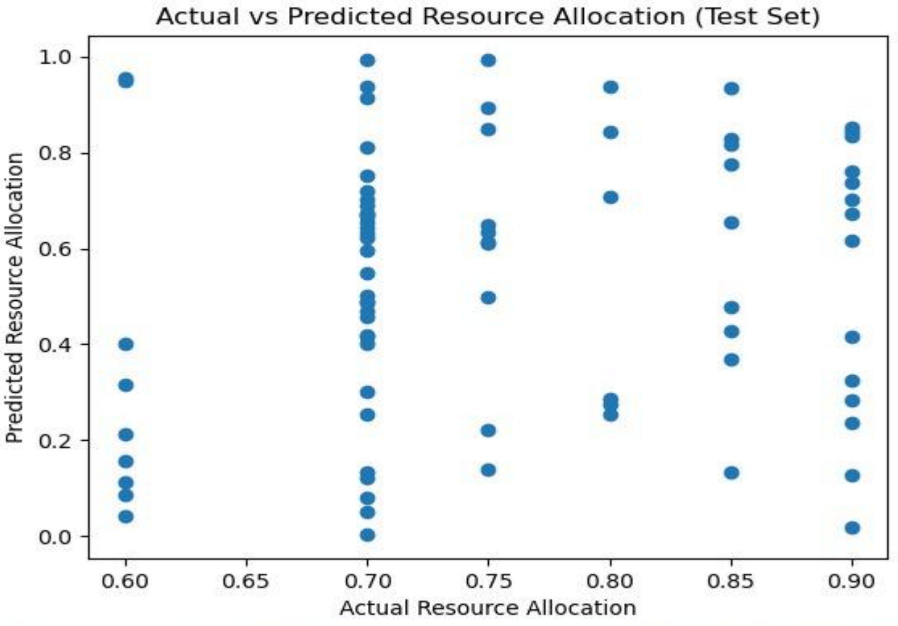
Genetic Algorithm Functions

Initialization of Population:
Initializes the population of VM-to-PM mappings with random assignments.
Parameters: Population size, number of virtual machines, number of physical machines.
Fitness Function:
Evaluates the fitness of each solution in the population based on burst time and turnaround time.
Higher fitness scores indicate better solutions.
Tournament Selection:
Selects parents for crossover using a tournament selection strategy.
Randomly selects individuals from the population and chooses the fittest individual as a parent.
Crossover:
Combines genetic information from selected parents to create offspring.
Randomly selects a crossover point and exchanges genetic material between parents to produce new solutions.
Mutation:
Introduces random variations into offspring to maintain genetic diversity.
Randomly selects genes in offspring and modifies them with a certain probability.
Genetic Algorithm for VM Placement:
Implements the genetic algorithm for virtual machine placement.
Iteratively evolves the population over multiple generations, applying selection, crossover, and mutation operators.
Returns the best solution from the final population

Genetic Algorithm Results

*

```
Dataset file found at: dataset\QOS_5G.csv
Genetic Algorithm-based Approach Metrics:
Mean Squared Error: 0.13029622646569147
Mean Absolute Error (MAE): 0.28553974757489287
Average Burst Time (Required Bandwidth): 0.32000000000000006 Mbps
Average Turnaround Time: 4.8913995 ms
Average Waiting Time: 4.5713995 ms
Average Actual Latency: 33.825 ms
Average Actual Bandwidth: 3.50238 Mbps
```



Random Forest Regression

Random Forest Regression (RFR):
RFR is a machine learning technique used for predictive modeling and forecasting.
It predicts resource demands in cloud computing by analyzing historical usage patterns and relevant factors.

Ensemble Learning:
RFR employs ensemble learning, where multiple decision trees make independent predictions.
The final prediction is determined by aggregating outputs from all trees.

Decision Trees:
Each tree is constructed by randomly selecting features and splitting the data.
This randomness reduces overfitting and improves generalization.

Predictive Power:
Trained on historical data, RFR predicts future resource demands based on input features like time, application type, and current utilization.Aggregated predictions offer robust resource requirement estimates.

Adaptability:
RFR adapts to changing data patterns.
The model can be retrained with new data to enhance prediction accuracy over time.

Random Forest Regression Methodology

Dataset Handling:
- The dataset is located within the designated folder using Python's `os` module.
- Dataset preprocessing involves reading CSV files and converting certain columns to appropriate data types.

Genetic Algorithm Functions:
- Several functions are defined for genetic algorithm operations, including:
  - Calculation of burst time and turnaround time based on VM-to-PM mapping.
  - Initialization of the population.
  - Fitness evaluation of each solution.
  - Tournament selection of parents.
  - Crossover and mutation operations on offspring.

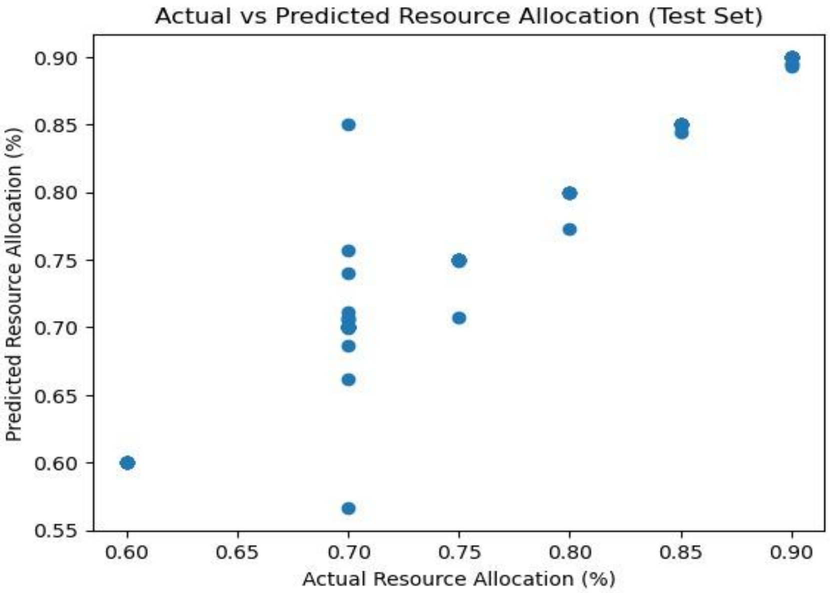
Model Training and Evaluation:
- The dataset is split into training and testing sets using `train_test_split`.
- The genetic algorithm is applied to train the model for virtual machine placement.
- Evaluation metrics such as mean squared error (MSE) and mean absolute error (MAE) are calculated.

Visualization:
- Scatter plots, line charts, and bar charts are generated to visualize actual vs predicted resource allocation.
- These visualizations aid in assessing the model's performance and understanding the allocation patterns.

Random Forest Regression Results

*

```
Dataset file found at: dataset\QOS_5G.csv
Mean Squared Error: 0.0006224218749999997
Mean Absolute Error (MAE): 0.0069812500000007
R-squared: 0.9289643796531586
Average Burst Time (Required Bandwidth): 2.8976499999999996 Mbps
Average Turnaround Time: 0.02198334375 ms
Average Waiting Time: -2.87566665625 ms
Average Actual Latency: 31.6625 ms
Average Actual Bandwidth: 3.2657249999999998 Mbps
```



Conclusion

In our exploration of resource allocation strategies in cloud computing, efficient data management and real-time processing emerge as critical factors for maximizing cloud service benefits. Our analysis underscores the profound impact of resource allocation methods on overall performance and customer satisfaction within cloud environments.

Insights from Comparison:
Comparing Random Forest Regression (RF) and Genetic Algorithm (GA) reveals compelling insights. RF demonstrates superiority with an impressive R-squared value of 0.9, surpassing GA's negative R-squared value. This emphasizes the importance of selecting methodologies tailored to specific resource allocation requirements.
Dynamic Nature of Cloud Computing:
Our study highlights the dynamic nature of cloud computing, evolving to meet user demands. Resource allocation, as the backbone of cloud infrastructure, demands swift and effective decision-making to ensure user satisfaction and provider profitability.
Future Directions:
Our findings pave the way for further exploration and refinement of resource allocation strategies. Future endeavors could leverage deep learning approaches to enhance problem-solving capabilities and performance analysis within cloud environments. Embracing technological advancements will drive innovation and deliver optimal solutions for resource allocation challenges.

References

Dataset :

https://www.kaggle.com/datasets/omarsobhy14/5g-quality-of-service

Research Paper :

- https://www.researchgate.net/publication/375084155_Resource_Allocation_in_Cloud_Computing

Thank You !!